S/N 09/982,794 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Li
Examiner: Divecha, Kamal B
Serial No.: 09/982,794
Group Art Unit: 2151
Filed: October 22, 2001
Docket No.: 0063-060001/BU1911
Title: DATA PATH OPTIMIZATION ALGORITHM

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

BRIEF ON APPEAL

(1) Real Party in Interest

Broadcom Corporation is the real party in interest.

(2) Related Appeals and Interferences

To the best of Appellant's knowledge, there are no related appeals or interference proceedings currently pending, which would directly affect or be directly affected by, or have a bearing on the Board's decision in this appeal.

(3) Status of Claims

Claims 1-13 are currently pending in the application, of which claims 1, 6, and 10 are independent claims. Appellant appeals the rejections of claims 1-13, which were finally rejected in the Final Office Action of July 11, 2008. Specifically, the Final Office Action of July 11, 2008 rejected claims 1-4, 6-8, and 10-12 under 35 U.S.C. §103(a) as being allegedly unpatentable as obvious over Thompson (European Publication No. EP 0572145 A2) ("Thompson") in view of Scott (U.S. Patent No. 6,512,773) ("Scott"), and further in view of Parruck, *et al.* (U.S. Patent No. 7,139,27 1) ("Parruck"). Claims 5, 9 and 13 were rejected under

35 U.S.C. 103(a) as being obvious over Thompson in view of Scott and further in view of Parruck and further in view of Yik (U.S. Patent No. 6,697,873) ("Yik").

(4) Status of Amendments

Claims 1, 6, and 10-13 were amended by virtue of Appellant's Response and Amendment to the Final Office Action of July 11, 2008, which was filed September 15, 2008. The amendments to claims 1, 6, and 10-13 were included to address objection(s) to those claims and/or rejection(s) of those claims under 35 U.S.C. §112(2). These amendments were entered by virtue of the Advisory Action of September 25, 2008, and were not otherwise addressed therein nor were the corresponding objections/rejections addressed therein. Therefore Appellant understands that the amendments to claims 1, 6, and 10-13 have been entered, and that the corresponding objections and rejections under 35 U.S.C. §112(2) have been withdrawn and are therefore moot for purposes of this Appeal. Consequently, the listing of the claims in the section (9) Claim Appendix below reflects the above understanding of the current status of the claims.

(5) Summary of Claimed Subject Matter

I. <u>Explanation</u>

The claimed subject matter relates to network switching for enabling high speed transfer of data through a network switching device. Such a device according to the invention accounts for otherwise disadvantageous situations in which a variable-length header is removed from one or more of the data cells, so that initially fixed-length data cells become variable length (and therefore slower and more difficult to switch/process). Although prior art systems and devices included techniques for mitigating the negative effects of such header removal(s), the presently-

claimed subject matter provides novel and non-obvious techniques for mitigating the effects of such header removal, to thereby obtain speed advantages associated with using fixed or constant-sized data cells. For example, in a conventional system, a destination network may have a backplane interface which supports forty-eight channels to accommodate cells having sixty-four bytes burst. However, the transmitting side of a source network may need to attach a four-byte header to the beginning of the packet. When the cells of the packet arrive at the destination network, the destination network may extract the four-byte header from the cell, thus leaving sixty bytes in the cell. This header removal step causes the cell to be four bytes short of the required format size since the cell now no longer satisfies the destination network's size requirement of sixty-four bytes. The system must wait for the next cell of the packet for this particular channel to arrive. The system, then, extracts the first four bytes of the next incoming cell, and combines the newly extracted bytes with the sixty bytes of the previous cell. Unfortunately, due to the misalignment of the cell after the removal of the header, the need to reconstruct the cells of the packet perpetuates throughout the transmission of all the subsequent cells of the packet. See Application, page 1, line 5, to page 5, line 13.

FIG. 1 illustrates an example of a non-limiting embodiment of the claimed subject matter in which an aggregator 100 is capable of eliminating such data misalignment in an advantageous manner. See Application, page 8, lines 1-3. In FIG. 1, cells 104 of the packet 106 enter an aggregator 100 via an uplink port 102 and are sent to a shared buffer 107 for storage and forwarding, as described in more detail below. See Application, page 18, lines 3-4. The aggregator 100 may be configured as a cell-based aggregator 100, wherein the data path structure of the aggregator 100 is optimized to accommodate, for example, a sixty-four-burst size. When the data packet 106 is received by the device 105, an ingress sub-module (not shown) may

remove the eight-byte header from the cell 104 containing the header (the header cell), leaving only fifty-six bytes remaining in the header cell of the packet 106. The aggregator 100 may remove the header of the packet so that the data of the packet may be processed. Then, the ingress sub-module of the device 105 may check to determine whether the cell 104 which included the header now contains a multiple of a predetermined number of bytes, for example, a multiple of the sixty-four-burst size. After the removal of the header, the remaining incoming cells 104 may be misaligned if the bytes contained in the header cell no longer satisfy a multiple of the predetermined number, as required by the aggregator's data structure. To prevent such data misalignment from occurring , the ingress sub-module of the device 105 may insert eight null-bytes into the header cell to replace the eight-byte header, which were removed. As a result of this null-byte insertion technique, this modified header cell will cause the remaining incoming cells to be aligned to the burst size mandated by the aggregator 100. The ingress sub-module of the channelized device 105 may also tag status information to the modified header cell to indicate the number of null-bytes inserted into the cell. Also as an ingress function, the ingress submodule of the channelized device 105 may determine the destination of the packet 106. The aggregator 100 may send the cells 104 to the shared buffer 107 so that the cells 104 are stored and positioned in the shared buffer 107 sequentially to re-assemble an assembled packet. Accordingly, when the shared buffer 107 receives a read signal, there is no need for the shared buffer 107 to rearrange and reassemble the cells of the packets sequentially. Since the cells 104 of the packets 106 have been assembled in the shared buffer 107 sequentially in this example(s), the aggregator 100 may not need to reassemble the cells 104 of the packet 106 before the packet is transmitted out of the aggregator 100. Then, as the cells 104 of the packet 106 exit the aggregator 100, another function of the egress sub-module is to determine if any cells of the

packet 106 contain null bytes by checking the cells to determine if any header cell was tagged with status information indicating that null bytes were inserted into the cell by the ingress submodule of the channelized device 105. If a header cell 104 of the packet 106 does contain null bytes, the egress module may read the status information and extract the null bytes from the header cell containing the null bytes. Namely, as the header cell 104 of the packet 106 exits the aggregator 100, the egress sub-module may strip the null bytes from the header cell and transfer the header cell and the remaining cells 104 of the packet 106 out of the aggregator 100. See Application, page 21, line 5 to page 24, line 16.

FIGS. 3A and 3B generally illustrate an example of a flow diagram of the handling of the cells 104 of a packet 106 when the cells 104 are received at an appropriate assembly line of the channelized device 105. In step 200, the port 102 may receive the packets from the source network. The cells may be arranged in sequential order at step 205. The ingress sub-module of the channelized device 105 may remove the header from the header cell and count the number of bytes in the extracted header in step 210. Then, in step 220, the ingress sub-module may count the bytes remaining in the header cell. In step 230, the ingress sub-module may check to determine whether the number of bytes remaining in the header cell is a multiple of a pre-determined number. If so, at step 235, the ingress module may transfer the cell to the buffer and return to step 200. If not, the process may advance to step 240 and may add null bytes to the header cell to replace the number of bytes contained in the extracted header. At step 240, the ingress module may also tag the cell with status information which indicates the number of null bytes inserted into the header cell. In step 250, the cells may then be transferred to the buffer memory. In step 255, the process may check to see if there are additional incoming cells. If so,

the process returns to step 200. If not, the ingress functions of the process may terminate in step 257. See Application, page 24, line 17 to page 25, line 12.

FIG. 3C generally illustrates an example of a flow diagram of the handling of the cells 104 of the packet 106 when the cells 104 are being transmitted from the aggregator 100. At the egress module (not shown) at step 260, the process may check to determine existence of a read request and thereafter readiness to receive the cells of a packet stored in the memory/buffer 107. If no, the process returns to step 260. If yes at step 260, the process may retrieve the cells associated with the requested packet and transfer the cells that make up the packet to the egress module at step 270. The egress module, in step 290, may check to determine whether any cells of the packet contain null bytes. If not, the process may proceed to step 300 and transfer of the cells to the buffer 107. If so, in step 290, the process may determine the number of nulls by reading the status information tag at step 310. The egress module may also remove the null bytes from the packet in step 310. Then, the egress module may transfer the packet out of the output port in step 300. See Application, page 25, line 13 to page 26, line 4.

II. Claim Charts with Specification Support

It is understood that the below are merely a few illustrative example embodiments to which the disclosed subject matter is not limited, and do not result in prosecution history estoppel and do not alter the scope of the claims.

1: A network device configured to prevent data misalignment of a data packet containing extra header bytes, the network device comprising::	
an ingress module having an input interface to receive a data packet comprising a plurality of cells, wherein a header cell of the data packet is one of the plurality of cells of the data packet, wherein the header cell of the plurality of cells comprises a header and packet data information and wherein the header cell includes the header in its entirety for the data packet	FIG. 1, elements 102, 104, 105, 106. See Application, page 18, lines 3-4; page 21, lines 5-20. Also FIG. 3A, elements 200, 205. Also see Application, page 24, lines 17-21.
a header detector configured to detect the header cell of the data packet and remove the header from the header cell of the data packet	FIG. 1, element 105. See Application, page 21, line 21 to page 22, line 7. Also FIG. 3A, element 210. Also see Application, page 24, lines 21-22.
a counter configured to determine whether the header cell of the data packet contains a multiple of a predetermined number of bytes after the header has been removed from the header cell;	FIG. 1, element 105. See Application, page 22, lines 7-13. Also FIG. 3A, element 210, 220, 230. Also see Application, page 24, line 22 to page 25, line 7.

an insertion module configured to insert null bytes into the header cell of the data packet to form a modified header cell of the data packet if the counter determines that the header cell of the data packet does not satisfy the multiple of the predetermined number of bytes in order to align all of a plurality of other cells of the packet	FIG. 1, element 105. See Application, page 22, lines 13-18. Also FIG. 3A, elements 230, 240. Also see Application, page 25, lines 4-12
an extraction module configured to remove the null bytes from the modified header cell of the data packet as a modified cell of the data packet exits the network device	FIG. 1, element . See Application, page 24, lines 6-16. Also FIG. 3C, elements 290, 310. Also see Application, page 25, line 13 to page 26, line 4.

6. A method of preventing data misalignment of a data packet containing extra header bytes, said method comprising::	
receiving, at an input port of a network device, a data packet comprising a plurality of cells, wherein a header cell of the data packet is one of the plurality of cells, wherein the header cell of the plurality of cells comprises a header and packet data information and wherein the header cell includes the header in its entirety for the data packet;	FIG. 1, elements 102, 104, 105, 106. See Application, page 18, lines 3-4; page 21, lines 5-20. Also FIG. 3A, elements 200, 205. Also see Application, page 24, lines 17-21.
detecting the header cell of the data packet;.	FIG. 1, element 105. See Application, page 21, line 21 to page

	22, line 7. Also FIG. 3A, element 210. Also see Application, page 24, lines 21-22.
removing the header from the header cell of the data packet;	FIG. 1, element 105. See Application, page 21, line 21 to page 22, line 7. Also FIG. 3A, element 210. Also see Application, page 24, lines 21-22.
determining whether the header cell of the data packet contains a multiple of a predetermined number of bytes after the header has been removed from the header cell;	FIG. 1, element 105. See Application, page 22, lines 7-13. Also FIG. 3A, element 210, 220, 230. Also see Application, page 24, line 22 to page 25, line 7.
inserting null bytes into the header cell of the data packet to form a modified header cell of the data packet if the counter determines that the header cell of the data packet does not satisfy the multiple of the predetermined number of bytes in order to align all of a plurality other cells of the packet	FIG. 1, element 105. See Application, page 22, lines 13-18. Also FIG. 3A, elements 230, 240. Also see Application, page 25, lines 4-12
forwarding the modified header cell of the data packet to an output port	FIG. 1, elements 100, 101, 107. See Application, page 23, line 18 to page 24, line 6.
removing the null bytes from the header cell of the data packet as a modified cell of the data packet exits the network device	FIG. 1, element . See Application, page 24, lines 6-16. Also FIG. 3C, elements 290, 310. Also see Application, page 25, line 13 to page 26, line 4.

10. A network device configured to prevent data misalignment of a data packet containing extra header bytes, the network device comprising:	
receiving means for receiving, at an input port of the network device, a data packet comprising a plurality of cells, wherein a header cell of the data packet is one of the plurality of cells of the data packet, wherein the header cell of the plurality of cells comprises a header and packet data information, and wherein the header cell includes the header in its entirety for the data packet;	FIG. 1, elements 102, 104, 105, 106. See Application, page 18, lines 3-4; page 21, lines 5-20. Also FIG. 3A, elements 200, 205. Also see Application, page 24, lines 17-21.
detecting means for detecting the header cell of the data packet;	FIG. 1, element 105. See Application, page 21, line 21 to page 22, line 7. Also FIG. 3A, element 210. Also see Application, page 24, lines 21-22.
header removing means for removing the header from the header cell of the data packet	FIG. 1, element 105. See Application, page 21, line 21 to page 22, line 7. Also FIG. 3A, element 210. Also see Application, page 24, lines 21-22.
determining means for determining whether the header cell of the data packet contains a multiple of a predetermined number of bytes after the header has been removed from the header cell	FIG. 1, element 105. See Application, page 22, lines 7-13. Also FIG. 3A, element 210, 220, 230. Also see Application, page 24, line 22 to page 25, line 7.

inserting means for inserting null bytes into the header cell of the packet to form a modified header cell of the data packet if the counter determines that the header cell of the data packet does not satisfy the multiple of the predetermined number of bytes in order to align all of a plurality of other cells of the packet	FIG. 1, element 105. See Application, page 22, lines 13-18. Also FIG. 3A, elements 230, 240. Also see Application, page 25, lines 4-12
forwarding means for forwarding the modified header cell of the data packet to an output port	FIG. 1, elements 100, 101, 107. See Application, page 23, line 18 to page 24, line 6.
null byte removing means for removing the null bytes from the modified header cell of the data packet as a modified cell of the data packet exits the network device	FIG. 1, element . See Application, page 24, lines 6-16. Also FIG. 3C, elements 290, 310. Also see Application, page 25, line 13 to page 26, line 4.

(6) Grounds of Rejection to be Reviewed on Appeal

The grounds of rejection to be reviewed on appeal are those grounds of rejection set forth in the July 11, 2008 Final Office Action. Specifically, the Final Office Action of July 11, 2008 rejected claims 1-4, 6-8, and 10-12 under 35 U.S.C. §103(a) as being allegedly unpatentable as obvious over Thompson (European Publication No. EP 0572145 A2) ("Thompson") in view of Scott (U.S. Patent No. 6,512,773) ("Scott"), and further in view of Parruck, *et al.* (U.S. Patent No. 7,139,27 1) ("Parruck"). Claims 5, 9 and 13 were rejected under 35 U.S.C. 103(a) as being obvious over Thompson in view of Scott and further in view of Parruck and further in view of Yik (U.S. Patent No. 6,697,873) ("Yik").

(7) Argument

I. Rejections under 35 U.S.C. § 103(a)

A. Claims 1-13

With reference first to claim 1, the Final Office Action of July 11, 2008 alleges that Thompson teaches all of the elements thereof, except for "a data packet including a plurality of cells, wherein a header cell of the data packet is one of the plurality of cells of the data packet and wherein the header cell of the plurality of cells comprises a header and packet data portion … and a counter to determine whether the header cell of the data packet contains a multiple of a predetermined number of bytes after the header has been removed from the header cell." See Final Office Action, page 13.

The Final Office Action instead relies on Scott and Parruck to make up for the admitted deficiencies of Thompson. Specifically, the Final Office Action alleges that Scott discloses "a counter to determine and/or count the number of octets of the user data PDU of the payload; and an insertion module that adds pad characters to make the frame or cell equal an integer number of 48 octet cells in order to align cells of the data packet…" and refers to Scott at column 10, lines 40-50 and to FIG. 5C, element 236 of Scott. See Final Office Action, page 13.

The Final Office Action goes on to admit that "Thompson in view of Scott does not disclose a data packet comprising a plurality of cells, wherein the header cell of the plurality of cells comprises a header and packet data portion and wherein the header cell includes the header in its entirety for the data packet." See Final Office Action, page 14. The Final Office Action goes on to allege that such a limitation is inherent to Scott, but nonetheless relies on Parruck to provide an allegedly explicit showing of the referenced claim language.

Applicant traverses the rejection of claim 1, because, as described in detail below, the cited references fail to disclose or render obvious all of the elements of claim 1. Moreover, Thompson explicitly teaches away from the proposed combination. Consequently, the Final Office Action of July 11, 2008 fails to establish a prima facie case of obviousness, so that the pending rejection of claim 1 should be withdrawn.

Claim 1 recites, in part, "a counter configured to determine whether the header cell of the data packet contains a multiple of a predetermined number of bytes after the header has been removed from the header cell" (similar claim recitations are also recited in independent claims 6 and 10, as referenced below with respect to those claims). As just described, the Final Office Action relies on column 10, lines 40-50 and FIG. 5C (e.g., element 236) of Scott as allegedly disclosing this feature(s) of claim 1. Applicant disagrees that Scott discloses determining whether the header cell of a data packet contains a multiple of a predetermined number of bytes after the header has been removed from the header cell, as recited in claim 1.

Column 10, lines 40-50 of Scott discloses (with emphasis added):

> "(i)n block 231, the payload (105a of FIG. 4A) is processed fiom the frame 100. The number of octets of the **user data** PDU 71 **of the payload is counted** in block 232. This value forms the length field of the AAL5 CS. **Note that the user data PDU 71 is the field found after the 4-octet ATM header field 91 of FIG. 4A.** Block 234 forms the W and CPI fields of the AAL5 frame. For the case where the UU and CPI field are not included in the header or trailer, the default "0" is used. Block 236 adds pad characters to make the AAL5 frame equal an integer number of 48 octet cells. In block 237, the 32 bit cyclic redundancy check (CRC) of the AAL5 frame is calculated. Block 238 segments the above AAL5 frame into an integer number of 48 octet cells."

As can be clearly observed from column 10, lines 40-50 of Scott the only part of the frame 100 that is "counted" is the "User Data PDU" (i.e,. 71) portion of the payload 105a. In

other words, the "4 octet ATM header" (i.e., 91) is <u>not</u> part of the counting operation performed by Scott. The only counting performed in Scott is counting on the number of octets of the user data included in the payload 71. The number of octets is used for the length field of the ATM adaptation layer-5 convergence sub-layer (AAL5 CS). The counting of the data octets in the payload is again reinforced in operation 232 of FIG. 5C. Once the segmentation of the 48 octet cells of data are prepared, then the 4 octet ATM header is extracted from the frame and HEC is added to create the 5 octet ATM header. The 5 octet ATM header is then combined with the 48 octet payload to create a conventional 53 octet ATM cell (operation 244).

Scott does not disclose using a counter to determine whether the header cell of the data packet contains a multiple of a predetermined number of bytes after the header has been removed from the header cell. The only operations that Scott performs after the header is extracted in operation 239 of FIG. 5C is adding the HEC to the header, adding the header back to the 48 octet payload, and appending a "last cell indicator" for the last cell (see operations 239-244 of FIG. 5C). Therefore, Scott does not disclose or suggest "a counter configured to determine whether the header cell of the data packet contains a multiple of a predetermined number of bytes after the header has been removed from the header cell," as recited, in part, in independent claim 1 and similarly in independent claims 6 and 10.

The Final Office Action of July 11, 2008 fails to address these deficiencies of Scott, and appears to acknowledge that Scott does not disclose the referenced language of claim 1, and to agree with Applicant's description of Scott, above. For example, the Final Office Action states, "Scott discloses … a counter to determine and/or count the number of octets of the user data PDU of the payload" (see Final Office Action, page 13) and "…Scott teaches and discloses a

counter that counts data octets of the user data PDU of the payload …" (See Final Office Action, page 14).

Furthermore, the Advisory Action of September 25, 2008 (i.e., lines 1-8 on page 2 thereof) also fails to address the fact that Scott does not disclose the missing elements of Thompson in the manner alleged by the Final Office Action. Specifically, the Advisory Action fails to address the fact that the specific elements of claim 1 are missing from Scott and therefore from the proposed combination of Scott with Thompson (and Parruck). Instead, the Advisory Action states (with emphasis added) that Scott "<u>is fully capable of</u> counting … the header cell of the packet…" However, this is no more than another admission that Scott does not, in fact, count the header cell of any packet disclosed therein, and therefore does not, in fact, disclose "a counter configured to determine whether the header cell of the data packet contains a multiple of a predetermined number of bytes after the header has been removed from the header cell," as recited in claim 1.

Indeed, the Final Office Action explicitly admits that Scott does not disclose or discuss the claimed header cell comprising a header and packet data information and including the header in its entirety for the data packet, as recited in claim 1. The Final Office Action alleges that such a feature is inherent in Scott, but then relies on Parruck to provide a showing of this claim element.

Without arguing or stipulating whether any such inherent showing of Scott exists, Applicant submits that it is inconsistent at best for the Final Office Action to take the position that a feature which is admittedly not even explicitly disclosed in Scott (e.g., the claimed "…header cell of the data packet contains a multiple of a predetermined number of bytes after the header has been removed from the header cell") is nonetheless manipulated thereby in the

manner suggested by the Final Office Action. That is, Applicant understands the possibility that a reference may include an inherent feature, but Applicant submits that such an (*arguendo*) inherent feature may not then be held to be, in this case, counted or otherwise used in the manner recited in claim 1.

In short, the Final Office Action appears not to appreciate the distinction between Scott (and Parruck) actually disclosing the admittedly missing elements of Thompson (which, as just shown, those references do not), as opposed to Scott (and Parruck) including elements (such as a counter, or a particular header type) which are merely "capable of" acting in a manner that would disclose the elements of claim 1 that are admittedly missing from Thompson if those elements were used according to Applicant's claimed invention.

In the latter case, a proper rejection would need to provide an explanation of why, for example, one of ordinary skill in the art would have used the counter of Scott in a manner that is admittedly not disclosed in Scott and in a way that the counter is, at best "capable of" being used. The present rejection falls short of any such showing. Rather, the present rejection merely states (with emphasis added) that "(o)ne of ordinary skill in the art would have been motivated because it would have determined and/or counted the number of bytes in a cell (Scott, col. 10 L40-50) and based on the determination it would have inserted the pad byte into the cell in order to align the headers and the cell (Thompson, col. 1 L25-38)." See Final Office Action, page 14.

Again, when referring to any relevant teaching of Scott with respect to "the cell," it must be understood from the above that the reference is to the data/payload portion, and not the header cell as in claim 1. Therefore, the rejection falls short of establishing a prima facie case of obviousness for at least this reason(s).

Further, as referenced above, the rejection fails further because Thompson explicitly teaches away the proposed combination. For example, in the portion of Thompson just referenced, Thompson discloses[1] a network adaptor that "…inserts at least one pad byte within one of the plurality of headers to cause the plurality of headers in the network packet to be aligned along predetermined multi-byte boundaries." See Thompson col. 1, lines 25-38. However, in the very next paragraph, Thompson further discloses "…based on the value of the destination service access point, the network adaptor places a number of pad bytes in the network link header." See Thompson, col. 1, lines 39-45. Thompson further discloses, "(n)etwork adaptor 12 searches the incoming byte stream for specific values in the destination … field of the network link header. The hardware will insert between 0 and 3 pad bytes … based on the value found in the destination SAP field." See Thompson, col. 4, L43-50.

Thus, Thompson specifically discloses that the insertion of pad bytes within a header(s) <u>is based on a destination address of the header</u>. Such a disclosure explicitly teaches away from the proposed modification of using the counter of Scott to "…determine whether the header cell of the data packet contains a multiple of a predetermined number of bytes after the header has been removed from the header cell." Applicant submits that even if (hypothetically) the counter of Scott were disclosed therein as being used in the manner of Applicant's claimed invention, modifying Thompson to use the counter of Scott thusly would render Thompson unsuitable for its intended purposes of adding pad bytes based on a destination field of the header.

---

[1] The Final Office Action references several alleged admissions made by the Applicant with respect to Thompson, with reference to Applicant's response of April 10, 2008 (See, e.g., Final Office Action, page 4). Applicant notes that any description of Thompson included in that response was intended merely to summarize the language of Thompson for the sake of convenience and brevity, and was not an attempt to characterize Thompson beyond what is explicitly described therein. Applicant therefore expressly disavows any portion of Applicant's previous response(s) which appear to characterize Thompson beyond what is explicitly stated therein.

For the sake of completeness, Applicant submits that Parruck fails to cure the deficiencies of Thompson and Scott with respect to the pending claims. Parruck discloses that a large ATM cell is divided into individual cells, where each of the individual cells include a header and a payload portion. Each data packet is divided into multiple header cells because each cell includes header and payload portions. There is no disclosure or suggestion in Parruck of "a counter configured to determine whether the header cell of the data packet contains a multiple of a predetermined number of bytes after the header has been removed from the header cell", as recited, in part, in independent claim 1 and similarly in independent claims 6 and 10.

Therefore, Applicants respectfully assert that the rejection under 35 U.S.C. §103(a) should be withdrawn because neither Thompson, Scott nor Parruck, whether taken singly or combined, teaches or suggests each feature of claims 1, 6 and 10. Each of claims 2-4, 7-8 and 11-12 depend on claims 1, 6 and 10 and should be allowed at least because of their dependence on claims 1, 6 and 10, in addition to the further limitations recited in claims 2-4, 7-8 and 11-12. The failure to teach each of the claim recitations of the pending claims demonstrates that a *prima facie* case of obviousness has not been established. Withdrawal of the rejections of claims 1-4, 6-8 and 10-12 is kindly requested.

As referenced above, claims 5, 9 and 13 were rejected under 35 U.S.C. 103(a) as being obvious over Thompson in view of Scott and further in view of Parmck and U.S. Patent No. 6,697,873 B 1 to Yik. According to the Final Office Action, Thompson, Scott and Parruck teach all of the elements of claims 5, 9 and 13 except for teaching that the medium access control protocol module has a MAC address for transmitting the modified cell of the data packet and a layer two switching module configured to build a table for forwarding rules upon which the

MAC address exists. Therefore, the Final Office Action combined the teachings of Yik with Thompson, Scott and Parruck to allegedly yield all of the elements of claims 5, 9 and 13.

The rejection is traversed as being based on references that neither teach nor suggest the novel combination of features clearly recited in independent claims 1, 6 and 10, upon which claims 5, 9 and 13 depend. Yik also does not cure the deficiencies of Thompson, Scott andlor Parruck, as outlined above. Yik teaches an apparatus and method for storing and searching computer node addresses in a computer network system. Each of claims 5, 9 and 13 depend on claims 1, 6 and 10 respectively, and thus, incorporates all of the elements of the independent claims. There is no teaching or suggestion in Yik of "a counter configured to determine whether the header cell of the data packet contains a multiple of a predetermined number of bytes after the header has been removed from the header cell", as recited, in part, in independent claim 1 and similarly in independent claims 6 and 10, upon which claims 5, 9 and 13 depend. Therefore, Applicants respectfully assert that the rejection under 35 U.S.C. §103(a) should be withdrawn because neither Thompson, Scott, Parruck nor Yik, whether taken singly or combined, teaches or suggests each feature of claims 1, 6 and 10. Each of claims 5, 9 and 13 depend on claims 1, 6 and 10 and should be allowed at least because of their dependence on claims 1, 6 and 10, in addition to the further limitations recited in claims 5, 9 and 13.

Thus, claims 1-13 recite subject matter which is neither disclosed nor suggested in the prior art references cited in the Office Action. It is therefore respectfully requested that all of claims 1-13 be allowed, and this application passed to issue.

(8) Conclusion

Appellant respectfully submits that all the pending claims, claims 1-13, in this application are patentable and requests that the Board of Patent Appeals and Interferences direct the Examiner to withdraw the rejections and move the application to allowance.

If necessary, please charge any additional fees or credit overpayment to Deposit Account No. 50-3521, referencing attorney docket number 0063-060001/BU1911.

Respectfully submitted,

/William G. Hughes, 46,112/

Dated: February 10, 2009

William G. Hughes
Reg. No. 46,112

Brake Hughes Bellermann LLP

202.470.6452 tel
202.470.6464 fax

(9) Claims Appendix

1. (Previously Presented) A network device configured to prevent data misalignment of a data packet containing extra header bytes, the network device comprising:

an ingress module having an input interface to receive a data packet comprising a plurality of cells, wherein a header cell of the data packet is one of the plurality of cells of the data packet, wherein the header cell of the plurality of cells comprises a header and packet data information and wherein the header cell includes the header in its entirety for the data packet;

a header detector configured to detect the header cell of the data packet and remove the header from the header cell of the data packet;

a counter configured to determine whether the header cell of the data packet contains a multiple of a predetermined number of bytes after the header has been removed from the header cell;

an insertion module configured to insert null bytes into the header cell of the data packet to form a modified header cell of the data packet if the counter determines that the header cell of the data packet does not satisfy the multiple of the predetermined number of bytes in order to align all of a plurality of other cells of the packet; and

an extraction module configured to remove the null bytes from the modified header cell of the data packet as a modified cell of the data packet exits the network device.

2. (Previously Presented) The network device as recited in claim 1 wherein the network device comprises an aggregator that interfaces with an Ethernet and a System Packet Interface Level 4 communication system.

3. (Previously Presented) The network device as recited in claim 2 wherein the aggregator is configured to interface between a twelve 1-Gigabit ports and one 12 Gigabitfs System Packet Interface Level 4 uplink.

4. (Original) The network device as recited in claim 2 comprises a network switch.

5. (Original) The network device as recited in claim 1 further comprising:

a medium access control (MAC) protocol module having a MAC address for transmitting the modified cell of the data packet; and

a layer two switching module configured to build a table of forwarding rules upon which the MAC addresses exist and to instruct the extraction module to remove the null bytes from the modified cell of the data packet as the modified cell of the data packet exits the network device.

6. (Previously Presented) A method of preventing data misalignment of a data packet containing extra header bytes, said method comprising:

receiving, at an input port of a network device, a data packet comprising a plurality of cells, wherein a header cell of the data packet is one of the plurality of cells,

wherein the header cell of the plurality of cells comprises a header and packet data information and wherein the header cell includes the header in its entirety for the data packet;

detecting the header cell of the data packet;

removing the header from the header cell of the data packet;

determining whether the header cell of the data packet contains a multiple of a predetermined number of bytes after the header has been removed from the header cell;

inserting null bytes into the header cell of the data packet to form a modified header cell of the data packet if the counter determines that the header cell of the data packet does not satisfy the multiple of the predetermined number of bytes in order to align all of a plurality other cells of the packet;

forwarding the modified header cell of the data packet to an output port; and

removing the null bytes from the header cell of the data packet as a modified cell of the data packet exits the network device.

7. (Previously Presented) The method as recited in claim 6, further comprising the step:

interfacing with an Ethernet and a System Packet Interface Level 4 communication system.

8. (Previously Presented) The method as recited in claim 7 wherein the interfacing occurs between a twelve 1-Gigabit ports and one 12-Gigabitls System Packet Interface Level 4 uplink.

9. (Original) The method as recited in claim 6 further comprising the steps of:

providing a medium access control (MAC) protocol module having a MAC address for transmitting the modified cell of the data packet; and

providing a layer two switching module configured to build a table of forwarding rules upon which the MAC addresses exist and to instruct the extraction module to remove the null bytes from the modified cell of the data packet as the modified cell of the data packet exits the network device.

10. (Previously Presented) A network device configured to prevent data misalignment of a data packet containing extra header bytes, the network device comprising:

receiving means for receiving, at an input port of the network device, a data packet comprising a plurality of cells, wherein a header cell of the data packet is one of the plurality of cells of the data packet, wherein the header cell of the plurality of cells comprises a header and packet data information, and wherein the header cell includes the header in its entirety for the data packet;

detecting means for detecting the header cell of the data packet;

header removing means for removing the header from the header cell of the data packet;

determining means for determining whether the header cell of the data packet contains a multiple of a predetermined number of bytes after the header has been removed from the header cell;

inserting means for inserting null bytes into the header cell of the packet to form a modified header cell of the data packet if the counter determines that the header cell of the data packet does not satisfy the multiple of the predetermined number of bytes in order to align all of a plurality of other cells of the packet;

forwarding means for forwarding the modified header cell of the data packet to an output port; and

null byte removing means for removing the null bytes from the modified header cell of the data packet as a modified cell of the data packet exits the network device.

11. (Previously Presented) The network device as recited in claim 10, further comprising the step:

interfacing with an Ethernet and a System Packet Interface Level 4 communication system.

12. (Previously Presented) The network device as recited in claim 11 wherein the interfacing occurs between a twelve 1-Gigabit ports and one 12-Gigabitls System Packet Interface Level 4 uplink.

13. (Previously Presented) The network device as recited in claim 10 further cornprising:

module providing means for providing a medium access control (MAC) protocol module having a MAC address for transmitting the modified cell of the data packet; and

table providing means for providing a layer two switching module configured to

build a table of forwarding rules upon which the MAC addresses exist and to instruct the extraction module to remove the null bytes from the modified cell of the data packet as the modified cell of the data packet exits the network device.

(10) Evidence Appendix

No separate evidence is submitted herewith.

(11) Related Proceedings Appendix

To the best of Appellant's knowledge, there are no related appeals or interference proceedings currently pending, which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal